# BlackBerry Java Application

Version: 5.0

## Feature and Technical Overview

BlackBerry

Published: 2010-04-12
SWD-789336-0412033819-001

# Contents


1 Overview ..... 3

2 New in this release ..... 4

Screen and UI ..... 4

Autocomplete text field ..... 4

Fields that appear temporarily on a screen ..... 4

Font customization ..... 5

Grid layouts ..... 5

Image creation ..... 6

Menu customization ..... 6

Picker dialog boxes ..... 6

Screen transitions ..... 8

Spin box ..... 9

Touch event injection ..... 10

Location-based services ..... 10

BlackBerry extension to JSR 179 ..... 10

Landmark information retrieval using geocoding and reverse geocoding ..... 11

Location picker dialog boxes ..... 11

Multimedia ..... 12

Media key event access ..... 12

Bitmap scaling ..... 13

Bitmap image comparison ..... 13

Image creation ..... 14

Image scrolling ..... 14

Image zooming and panning ..... 14

Support for high-performance graphics by using OpenGL ES ..... 15

Video capture ..... 15

Personal Information Management ..... 15

Multiple contact lists support ..... 15

Linking a contact ..... 16

Combined PIM lists support ..... 16

BlackBerry Browser ..... 16

Enhanced support for web content in BlackBerry device applications ..... 16

Messages ..... 17

Automatic download of email attachments 17
Phone 17
Message waiting indicator resets 17
Multi-line phone support 17
Phone screen customization 18
MEID retrieval 19
Support for the BlackBerry Push Service on a BlackBerry device 19
JavaScript extensions for a BlackBerry Widget application 19
SQLite 19
Connections 20
Enhanced Network API 20
Hotspot API 20
Security 21
Biometric recognition and authentication 21
User authentication modules 21
BlackBerry Java Application packaging and distributing 21
Location choices for adding your application on a BlackBerry device 21

3 System requirements 22

4 Related resources 23

5 Glossary 24

6 Provide feedback 26

7 Document revision history 27

8 Legal notice 29

# Overview

1

BlackBerry® APIs provide you with the classes and interfaces that you can use to create a custom application with features that are designed specifically for a BlackBerry device. You can use the BlackBerry APIs to create applications that perform the following functions:

- interact with specific built-in features of a BlackBerry device, such as the GPS and mapping functions, the camera and video recorder, the phone, and peripherals that use Bluetooth® technology
- respond to interaction with the input methods on the BlackBerry device, such as the trackball, touch screen, or keyboard
- integrate with BlackBerry® Device Software applications, such as the contacts application, the messages application, the BlackBerry® Browser, and BlackBerry® Maps
- interact with the components of the BlackBerry solution, including the BlackBerry® Enterprise Server, push services, and server-side resources

# New in this release

2

## Screen and UI

### Autocomplete text field

You can help users to type data more quickly, by using the `AutoCompleteField` class that is provided in the `net.rim.device.api.ui.component` package to create an autocomplete text field. You can display the autocomplete text field on the screen of the BlackBerry® device. As the user types in the field, the autocomplete text field searches against a set of data and displays items that match what the user typed. You can specify the data that the typed string is compared against. The data can be a set of literal strings or specified fields from a support data source on the BlackBerry device.

For example, you can create an autocomplete text field that enables a user to quickly type a contact's name by typing part of the name or birthday of the contact that is available on the BlackBerry device. You can specify the words or phrases to display to the user as the matches occur. For example, you can display the name, phone number, and email address of contacts in the returned results.

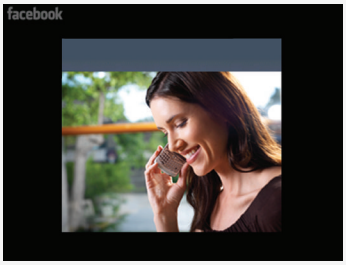
To download a sample application that demonstrates how to use autocomplete text fields, visit www.blackberry.com/go/autocompletefieldsample. For more information about autocomplete text fields, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

### Fields that appear temporarily on a screen

You can display fields on a pair of temporary managers that appear on the top and bottom of a screen by using the `EyelidFieldManager` class that is provided in the `net.rim.device.api.ui.extension.container` package. For example, when users view an image, you can display a description of the image along the top or bottom of the screen. You can define the amount of time that the information appears on a screen.

By default, the fields appear when the user moves the trackball or, on a BlackBerry® device with a touch screen, when the user touches the screen. The fields disappear after a period of inactivity.

To download a sample application that demonstrates how to use temporary fields, visit www.blackberry.com/go/eyelidfieldsample. For more information about temporary fields, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.




Before information displays on a screen temporarily

When information displays on a screen temporarily

## Font customization

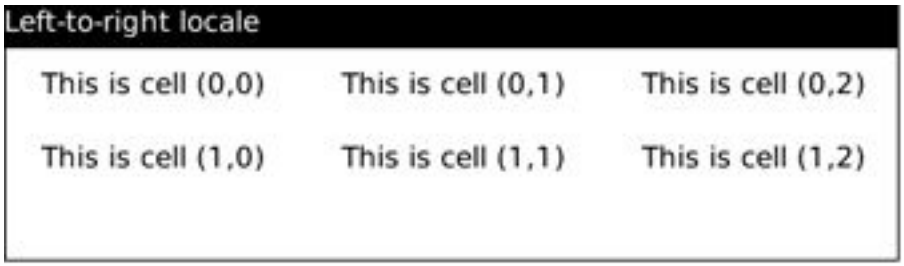
You can install fonts on, and remove fonts from, a BlackBerry® device by using the `FontManager` class that is provided in the `net.rim.device.api.ui` package. For example, you can install a font on a device, use the font in your application, and enable other applications to use the font.

For more information about fonts, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Grid layouts

You can position fields in rows and columns on a screen to create a grid by using the `GridFieldManager` class that is provided in the `net.rim.device.api.ui.container` class. After you create a grid, you can add fields, delete fields, and specify the grid's properties. For example, you can use a grid layout to display multiple images on a screen.

For more information about grid layouts, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.




## Image creation

You can add an image to your application from a bitmap or an encoded image by using the Image API that is provided in the `net.rim.device.api.ui.image` package. For example, you can use an `Image` as an icon in a menu item.

For more information about the Image API, see the API reference for the BlackBerry® Java® Development Environment.

## Menu customization

You can customize a menu in your BlackBerry® device application.

| Customization | Description |
|---|---|
| background, border, and font | You can specify a custom background, border, and font for a menu in your application by using the `Menu.setBackground`, `Menu.setBorder`, and `Menu.setFont` methods in the `net.rim.device.api.ui.component` package. |
| menu icons | You can add an icon to a menu item in your BlackBerry device application by using the `MenuItem.setIcon` method in the `net.rim.device.api.ui` package. |

To download a sample application that demonstrates how to customize a menu, visit www.blackberry.com/go/menusample. For more information on customizing menus, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

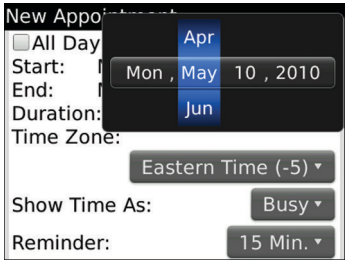
## Picker dialog boxes

You can use picker dialog boxes to help users choose an item from a list.

## Date and time picker dialog boxes

You can enable users to choose a day, month, year, or time, by using the `DateTimePicker` class that is provided in the `net.rim.device.api.ui.picker` package to create a date and time picker dialog box. For example, you can enable users to choose a month and a year to specify the expiry date for a credit card.

To download a sample application that demonstrates how to use date and time pickers, visit www.blackberry.com/go/spinnersample. For more information about date and time pickers, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

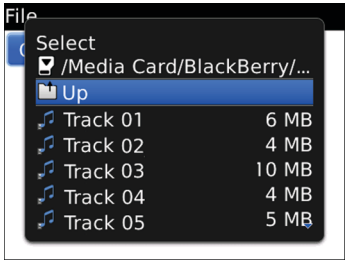



**Related topics**
Spin box, 9

## File picker

You can enable users to navigate to the files on a BlackBerry® device, by using the `net.rim.device.api.ui.picker.FilePicker` class to create a file picker. For example, you can enable users to search for a song on the BlackBerry device or on a media card.

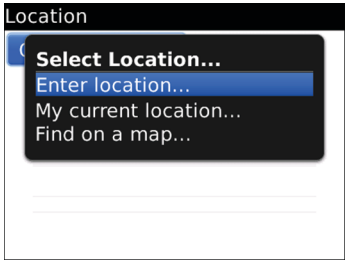
To download a sample application that demonstrates how to use file pickers, visit www.blackberry.com/go/filepickersample. For more information about file pickers, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.



### Location picker dialog boxes

You can enable users to choose a location from a list that you define, by using the `LocationPicker` class that is provided in the `net.rim.device.api.lbs.picker` package to create a location picker dialog box. For example, you can enable users to choose their current GPS location or a previous location.

To download a sample application that demonstrates how to use location pickers, visit www.blackberry.com/go/locationpickersample. For more information about location pickers, see the API reference for the BlackBerry® Java® Development Environment.

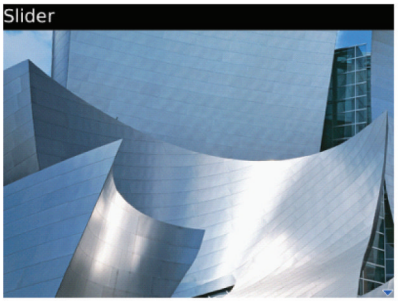



## Screen transitions

You can apply a visual effect that appears when your application opens or closes a screen on a BlackBerry® device by using the `TransitionContext` class in the `net.rim.device.api.ui` package. For example, you can reveal a new screen that your application opens by sliding it onto the display on the device. After you create a screen transition, you must register it within your application, specify the outgoing and incoming screens, the events that cause the transition to occur, and the transition to display using methods available in the `UiEngineInstance` class.

**Note:** The theme on the BlackBerry device controls the visual effects that display when a user opens an application. For more information, see the *BlackBerry Theme Studio User Guide*.

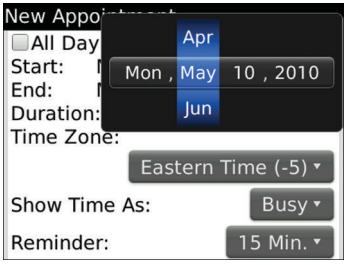
To download a sample application that demonstrates how to use screen transitions, visit www.blackberry.com/go/screentransitionssample. For more information on screen transitions, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.




## Spin box

You can allow users to choose an item from an ordered list, by using the `SpinBoxField` class that is provided in the `net.rim.device.api.ui.component` package to create a spin box. For example, you can create a spin box that allows users to find a number or to change the day of the week. A date and time picker dialog box is an example of a spin box.

For more information about spin boxes, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.




**Related topics**

Date and time picker dialog boxes, 7

## Touch event injection

On BlackBerry® devices with touch screens, you can inject a touch event, such as the slide or tap of a finger on the screen, by using the `EventInjector.TouchEvent` class available in the `net.rim.device.api.system` package. For example, you can use touch event injection to automate the testing of your application or to allow interaction between a peripheral and a BlackBerry device.

To download a sample application that demonstrates how to use touch event injection, visit www.blackberry.com/go/toucheventinjectorsampleapp. For more information about the sample application, visit www.blackberry.com/go/devguides to read the *Touch Event Injector Sample Application Overview*. For more information about touch event injection, see the *BlackBerry Java Application UI and Navigation Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

# Location-based services

## BlackBerry extension to JSR 179

You can access GPS location information about a BlackBerry® device by using the BlackBerry extension to JSR 179 that is available in the `net.rim.device.api.gps` package. You can use the BlackBerry extension to JSR 179 on BlackBerry devices that run BlackBerry® Device Software 5.0 or later.

| Class | Description |
|---|---|
| `BlackBerryCriteria` | The `BlackBerryCriteria` class extends the `Criteria` class and permits you to specify the following information:<br>• initial GPS mode<br>• failover GPS mode, if applicable<br>• GPS mode to use after your application obtains a GPS fix |
| `BlackBerryLocation` | The `BlackBerryLocation` class extends the `Location` class and permits you to retrieve the following information:<br>• GPS satellite information (for example, the number of GPS satellites, the average signal quality, and details about the GPS satellites)<br>• data source that obtains the GPS fix<br>• GPS mode that provides the GPS location information |

| Class | Description |
| --- | --- |
| `BlackBerryLocationProvider` | The `BlackBerryLocationProvider` class extends the `LocationProvider` class and permits you to perform the following actions:<br>• process a GPS location request from a `BlackBerryCriteria` object<br>• pause and resume a location listener<br>• retrieve the GPS provider type (for example, an internal GPS receiver or a Bluetooth® enabled GPS receiver) |

For more information about the BlackBerry extension to JSR 179, see the *BlackBerry Java Application Location-Based Services Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Landmark information retrieval using geocoding and reverse geocoding

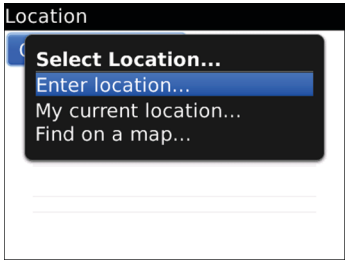
You can retrieve landmark information by invoking the `Locator.geocode()` and `Locator.reverseGeocode()` methods in the `net.rim.device.api.lbs` package. For example, you can use geocoding to find landmark information for a known landmark by using a street address, city, or postal code. You can use reverse geocoding to find an approximate street address by using the latitudinal and longitudinal coordinates.

For more information about geocoding and reverse geocoding, see the *BlackBerry Java Application Location-Based Services Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Location picker dialog boxes

You can enable users to choose a location from a list that you define, by using the `LocationPicker` class that is provided in the `net.rim.device.api.lbs.picker` package to create a location picker dialog box. For example, you can enable users to choose their current GPS location or a previous location.

To download a sample application that demonstrates how to use location pickers, visit www.blackberry.com/go/locationpickersample. For more information about location pickers, see the API reference for the BlackBerry® Java® Development Environment.




# Multimedia

Enhancements to the BlackBerry® APIs enables you to embed a viewfinder, record video clips, and play back captured video. You can also create a BlackBerry device application that can respond to key presses of media keys even when the application is in the background. New methods are also available to compare `Bitmap` images to find partial differences, and to scale `Bitmap` images.

## Media key event access

You can access media key events on a BlackBerry® device by using the Media Key API that is provided in the `net.rim.device.api.media` package. For example, when a user presses a volume key on the BlackBerry device, your application can respond and increase or decrease the volume accordingly.

You can access media key events while your application is in the background by implementing the `MediaActionHandler` interface and registering your application by invoking `Application.addMediaActionHandler()`. For example, your application can respond when the user presses the Mute key while the user is using another application on the BlackBerry device.

Your application can access media key events only if the application on the top of the application stack is not registered to listen for the media key events. If multiple applications that are running in the background are registered to listen for media key events, the application that is closest to the top of the application stack receives the media key event.

MIDlet applications can access media key events by using extensions that are provided in the `net.rim.device.api.lcdui.control` package.

To download a sample application that demonstrates how to use the Media Key API, visit www.blackberry.com/go/mediaplayerdemo. For more information about media keys, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Bitmap scaling

You can scale bitmap images by using the `Bitmap.scaleInto()` method that is provided in the `net.rim.device.api.system` package. The method creates a scaled copy of the original bitmap image. When you scale a bitmap image, you can specify a filter and whether to maintain the aspect ratio of the original image. The filter controls the interpolation of the scaled bitmap image.

The following table describes the filtering options and illustrates the effects of the filter on a scaled image.

| Filter | Description |
|---|---|
| `FILTER_BOX` | This filter has fast performance but provides a bitmap image of low quality.<br> |
| `FILTER_BILINEAR` | This filter has performance that is slower than the `FILTER_BOX` filter but provides an image of better quality.<br> |
| `FILTER_LANCZOS` | This filter has slow performance but provides a bitmap image of high quality.<br> |

For more information about bitmap scaling, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Bitmap image comparison

You can compare bitmap images by using the `Bitmap.locateDifference()` method that is provided in the `net.rim.device.api.system` package. For example, you can compare two versions of an image changes that were made to the image. After you identify the changes, you can render only the parts that changed and avoid repainting a screen unnecessarily.

For more information about comparing bitmap images, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Image creation

You can add an image to your application from a bitmap or an encoded image by using the Image API that is provided in the `net.rim.device.api.ui.image` package. For example, you can use an `Image` as an icon in a menu item.

For more information about the Image API, see the API reference for the BlackBerry® Java® Development Environment.

## Image scrolling

You can display a row of images on a BlackBerry® device screen that a user can scroll through, by using the `PictureScrollField` class that is provided in the `net.rim.device.api.ui.extension.component` package. Users can scroll through the images by rolling the trackball, pressing shortcut keys, or, on BlackBerry devices with a touch screen, sliding a finger across the screen. For example, you can create a field that displays a row of images, and for each image you can provide a caption and tooltip that appears when the user scrolls through the images.

To download a sample application that demonstrates how to use the `PictureScrollField` class, visit www.blackberry.com/go/picturescrollfieldsample. For more information about the `PictureScrollField` class, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Image zooming and panning

You can enable users to zoom into and pan an image by using the `ZoomScreen` class that is provided in the `net.rim.device.api.ui.extension.container` package. For example, when a user clicks the trackball, trackpad, or touch screen, the BlackBerry® device zooms into the center of the image. To pan the image, the user can zoom into the image, and then roll the trackball or slide a finger on the trackpad or across the screen. To zoom out, the user can press the Escape key.

To download a sample application that demonstrates how to use image zooming and panning, visit www.blackberry.com/go/zoomscreensample. For more information about image zooming and panning, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Support for high-performance graphics by using OpenGL ES

You can create 2-D and 3-D graphics for your BlackBerry® device applications by using JSR 239 and BlackBerry APIs. JSR 239 contains the Java® binding for OpenGL® ES. OpenGL ES is used to create graphics that are displayed on wireless devices.

The following packages provide support for OpenGL ES:

- `net.rim.device.api.math`
- `net.rim.device.api.opengles`
- `javax.microedition.khronos.egl`
- `javax.microedition.khronos.opengles`
- `java.nio`

For more information about OpenGL ES, visit www.khronos.org/opengles/. For more information about support for OpenGL ES in BlackBerry device applications, see the *BlackBerry Java Application Multimedia Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Video capture

You can use the camera application on the BlackBerry® device to record videos by using JSR 135 and the `Manager` class that is provided in the `javax.microedition.media` and `javax.microedition.media.control` packages. For example, you can create an application that records videos and manages the viewfinder of the camera application.

You can record the video to a .3GP file on the BlackBerry device or record the video to an output stream. Data is written to the output stream in a proprietary format.

To download a sample application that demonstrates how to record a video, visit www.blackberry.com/go/videorecordingsample. For more information about video recording, see the *BlackBerry Java Application Multimedia Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

# Personal Information Management

## Multiple contact lists support

Your BlackBerry® device application can access multiple contact lists on a BlackBerry device by using the `PIM` class that is provided in the `javax.microedition.pim` package. For example, your application can access the contact list in the contacts application on the BlackBerry device, and the contact list in other applications on the BlackBerry device.

Your application can access all the contact lists on the BlackBerry device by invoking `PIM.listPIMLists (PIM.CONTACT_LIST)`. Support for access to multiple contact lists was introduced in BlackBerry® Java® Development Environment 5.0. In earlier versions, your application could access only the active contact list by invoking this method. You can identify each contact list on the BlackBerry device by a name or by a UID.

For more information about access to multiple contact lists, see the *BlackBerry Java Application Integration Development Guide* and the API reference for the BlackBerry Java Development Environment.

## Linking a contact

You can link a contact in your BlackBerry® Java® Application with a contact in the contacts application on the BlackBerry device by using the classes and interfaces in the `net.rim.blackberry.api.pdap.contactlinking` package.

For example, you can allow BlackBerry device users to link a contact in your CRM application with a contact in the contacts application on the BlackBerry device. When a user views a linked contact in the contacts application, you can display contact information that is from your application.

You can create menu items that appear in the contacts application when a user views a contact that is linked with a contact in your application. The menu items can perform any action you want. For example, you can allow a user to add notes in the contacts application about the contact in your application.

## Combined PIM lists support

You can access a PIM list that combines multiple lists, either multiple contact lists or multiple calendars, that are on a BlackBerry® device. To access a combined list, invoke one of the `BlackBerryPIM.openUnifiedPIMList()` methods.

For more information, see the API reference for the BlackBerry® Java® Development Environment.

# BlackBerry Browser

## Enhanced support for web content in BlackBerry device applications

In BlackBerry® Java® Development Environment 5.0, the `net.rim.device.api.browser.field2` package provides a new set of APIs you can use to embed web content in BlackBerry device applications.

In earlier versions of the BlackBerry JDE, the `net.rim.device.api.browser.field` package provided functionality to add web content to a BlackBerry device application. For more information about using this browser field, see the BrowserFieldDemo sample application that is provided with the BlackBerry JDE.

The APIs that are new in BlackBerry JDE 5.0 allow you to perform the following actions:

- load and configure the display settings for web content in `Field` objects in any application

- access the DOM for loaded web content
- control the HTTP handling for connections, SSL, cookies, and caching
- set callback listeners to monitor the loading progress of web pages

You can create applications that include Java objects and methods that can access and invoke JavaScript® code and be accessed and invoked by JavaScript code by using the APIs in the `net.rim.device.api.script` package, which is new in BlackBerry JDE 5.0.

# Messages

## Automatic download of email attachments

You can automatically download attachments in incoming email messages when messages arrive in the BlackBerry® device message list by using the `AttachmentDownloadManager` class that is provided in the `net.rim.blackberry.api.mail` package. You can determine the status of the download by implementing the `DownloadProgressListener` interface.

For more information about downloading email attachments, see the *BlackBerry Java Application Integration Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

# Phone

## Message waiting indicator resets

You can check for and remove the message waiting indicator on a BlackBerry® device by using the `Phone` class in the `net.rim.blackberry.api.phone` package. For example, you can check if the BlackBerry device is displaying the message waiting indicator and remove the indicator.

For more information about checking and removing the message waiting indicator on a BlackBerry device, see the API reference for the BlackBerry® Java® Development Environment.

## Multi-line phone support

A BlackBerry® device can support multiple types of phone lines, including the cellular lines that are provided by a wireless service provider and work lines that are integrated with an organization's PBX phone system. You can create a BlackBerry device application that interacts with the phone lines that are available on a BlackBerry device by using the methods that are provided

in the `net.rim.blackberry.api.phone.Phone` class and by using the `net.rim.blackberry.api.phone.MultiLineListener` class. For example, you can create an application that verifies the types of phone lines that are available, and uses a specific phone line to make an outgoing call.

To download a sample application that demonstrates how to use multiple phone lines, visit www.blackberry.com/go/phonemultilinesample. For more information about developing an application that uses phone line information, see the *BlackBerry Java Application Integration Development Guide* and the API reference for the BlackBerry Java Development Environment.

## Phone screen customization

You can customize the phone screens on a BlackBerry® device to display content about a caller by using the `PhoneScreen` class that is provided in the `net.rim.blackberry.api.phone.phonegui` package. For example, you can display content that is provided by your CRM application about a customer.

The device displays the content on the lower part of the phone screen. If the device has multiple applications that can provide information about a caller, the device displays content from an application for approximately two seconds before displaying content from the next application.

To download a sample application that demonstrates phone screen customization, visit www.blackberry.com/go/phonescreensample. For more information, see the *BlackBerry Java Development Environment Integration Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## MEID retrieval

An MEID is a unique ID that identifies a mobile device that operates on a CDMA network. A BlackBerry® device that operates on a CDMA network has an MEID. You can retrieve the MEID by using the `CDMAInfo.getHexMEID()` and `CDMAInfo.getDecimalMEID()` methods that are provided in the `net.rim.device.api.system` package.

For more information about retrieving an MEID, see the API reference for the BlackBerry® Java® Development Environment.

# Support for the BlackBerry Push Service on a BlackBerry device

You can implement the features of the BlackBerry® Push Service SDK on a BlackBerry device by using the Push API that is provided in the `net.rim.blackberry.api.push` package. The Push API enables you to create a push application that can perform the following functions:

- register your application with the BlackBerry® Push Service
- listen for push notifications sent to the BlackBerry device
- start your application when the BlackBerry device receives a push notification from your application
- send a notification after a specified event occurs on the BlackBerry device (for example, after a user changes the SIM card in the BlackBerry device)
- send a notification when a user removes your application from the BlackBerry device

For more information about the BlackBerry Push Service, visit www.blackberry.com/developers/pushservice. For more information about the Push API, see the API reference for the BlackBerry® Java® Development Environment.

# JavaScript extensions for a BlackBerry Widget application

You can register your own JavaScript® extensions with your BlackBerry® Widget application by using the `WidgetExtension` interface that is provided in the `net.rim.device.api.web` package. For example, you can create your own JavaScript extension that can expose BlackBerry device capabilities that the BlackBerry Widget API does not expose.

For more information about developing a BlackBerry Widget application, see the *BlackBerry Widget SDK Development Guide*, the API reference for the BlackBerry® Widget SDK, and the API reference for the BlackBerry® Java® Development Environment.

# SQLite

You can create an application that uses the SQLite® database library that is available on a BlackBerry® device. SQLite is an open source relational database library. BlackBerry devices that run BlackBerry® Device Software 5.0 or later include the SQLite library.

SQLite is designed to make efficient use of memory resources and includes only a small set of features. As a result, it can be a good choice for embedded and wireless applications. You should consider using the SQLite API if you want to store application data, and your application would benefit from the additional structure and manageability that a relational database can provide.

You can use the SQLite API, provided in the `net.rim.device.api.database` package, to store application data persistently to eMMC memory or a microSD card. The SQLite API provides the option to store data in an encrypted format using the same mechanisms as the Persistent Store APIs.

To download a sample application that demonstrates how to use the SQLite API, visit www.blackberry.com/go/sqlitedemo. For more information about SQLite, see the *BlackBerry Java Application SQLite Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

# Connections

## Enhanced Network API

You can specify how your application opens a connection to a wireless network by using the enhanced Network API that is provided in the `net.rim.device.api.io.transport` and `net.rim.device.api.io.transport.options` packages. The Network API simplifies how your application establishes a connection to a wireless network and enables you to find the wireless transport types available on the BlackBerry® device. For example, you can use the Network API to find the availability and signal strength of a Wi-Fi® network that your application can use to open an HTTP connection.

You can use the Network API to specify a preferred list of wireless transport types. For example, if a Wi-Fi network is not available or does not have sufficient signal strength, you can have your application search for other transport types, such as a cellular network. If you do not specify a transport type, by default the BlackBerry device searches for all available transport types.

For more information about the Network API, see the *BlackBerry Java Application Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## Hotspot API

You can create an application that helps users connect to hotspots, by using the Hotspot API that is provided in the `net.rim.device.api.wlan.hotspot` package. Your application can interact with the Wi-Fi® transceiver on a Wi-Fi enabled BlackBerry® device, and the application that users use to manage a connection to a Wi-Fi network.

Previously, to connect to a hotspot a user completed multiple tasks. Now, to automate the process of associating and authenticating a user to a hotspot, you can use the Hotspot API to create an application that performs the following functions:

- automatically associate a Wi-Fi profile on the device, with a hotspot
- automatically log in users to a hotspot
- store a user's hotspot login information on the device
- integrate with the setup wizard and the Wi-Fi profiles that users use to manage a connection to a Wi-Fi network

For more information about the Hotspot API, see the API reference for the BlackBerry® Java® Development Environment.

# Security

## Biometric recognition and authentication

You can store, manage, and retrieve fingerprint information on a BlackBerry® device by using the Biometric API that is provided in the `net.rim.device.api.biometrics` package and the methods that are provided in the `SmartCard` class and `SmartCardSession` class. For example, you can retrieve fingerprint information from a smart card and compare the information with a scanned fingerprint.

For more information about the Biometric API and the Smart Card API, see the *BlackBerry Java Application Cryptographic Smartcard Driver Development Guide* and the API reference for the BlackBerry® Java® Development Environment.

## User authentication modules

You can create a module that the security infrastructure on a BlackBerry® device can use to authenticate users by using the `net.rim.device.api.userauthenticator` package. For example, you can create a module that enables the use of biometric hardware to authenticate a user to the BlackBerry device. Custom user authentication modules are designed to add additional authentication to the standard password authentication available on the BlackBerry device.

For more information about creating user authentication modules, see the API reference for the BlackBerry® Java® Development Environment.

**Related topics**
Biometric recognition and authentication, 21

# BlackBerry Java Application packaging and distributing

## Location choices for adding your application on a BlackBerry device

You can add your BlackBerry® device application to an application folder that is available with the theme on the BlackBerry device, to a new application folder that you create, or to the Home screen on the BlackBerry device. For example, if you develop a game, you can add your game to the Games folder on the device. If you do not specify a location, the device adds your application to the default application folder (for example, the Downloads application folder). You can specify the location to add your application to on devices that are running BlackBerry® Device Software 5.0 or later.

You can specify the location to add your application to by adding the `Content-Folder` attribute to the .jad file for your application. For more information, see the *BlackBerry Java Application Development Guide*.

# System requirements

3

| Item | Requirement |
| --- | --- |
| development environment | One of the following development environments:<br>• BlackBerry® Java® Development Environment<br>• BlackBerry® Java® Plug-in for Eclipse® 1.1 or later with BlackBerry® Java® SDK 5.0<br>• another third-party development environment, such as NetBeans™, that is running BlackBerry Java SDK Component Package 5.0 |
| BlackBerry Smartphone Simulator or BlackBerry device | Any BlackBerry Smartphone Simulator or device running BlackBerry® Device Software 5.0 or later. |

# Related resources

4

To download tools that you can use to create, test, package, and distribute applications for the BlackBerry® device, visit www.blackberry.com/developers/downloads.

For developer guides, visit www.blackberry.com/go/devguides.

For more information about BlackBerry APIs, see the API reference for the BlackBerry® Java® Development Environment by visiting www.blackberry.com/go/apiref.

# Glossary

5

**2-D**
two-dimensional

**3-D**
three-dimensional

**AES**
Advanced Encryption Standard

**API**
application programming interface

**CDMA**
Code Division Multiple Access

**CRM**
customer relationship management

**DOM**
Document Object Model

**DTMF**
Dual Tone Multiple-frequency

**GPS**
Global Positioning System

**GSM**
Global System for Mobile communications®

**HTTP**
Hypertext Transfer Protocol

**HTTPS**
Hypertext Transfer Protocol over Secure Sockets Layer

**JSR**
Java® Specification Request

**MEID**
Mobile Equipment Identifier

**MIDP**
Mobile Information Device Profile

**PBX**
Private Branch Exchange

**PIN**
personal identification number

**POP**
Post Office Protocol

**RMS**
Record Management System

**SIM**
Subscriber Identity Module

**SMS**
Short Message Service

**SQL**
Structured Query Language

**SSL**
Secure Sockets Layer

**SVG**
Scalable Vector Graphics

**TCP**
Transmission Control Protocol

**Triple DES**
Triple Data Encryption Standard

**UID**
unique identifier

# Provide feedback

6

To provide feedback on this deliverable, visit www.blackberry.com/docsfeedback.

# Document revision history

7

| Date | Description |
|---|---|
| 12 April 2010 | Changed the Touch event injection topic. |
| 6 April 2010 | Added the following topics:<br>• Combined PIM lists support<br>• Hotspot API<br>Changed the following topics:<br>• Bitmap scaling<br>• Fields that appear temporarily on a screen<br>• Image scrolling<br>• JavaScript extensions for a BlackBerry Widget application<br>• Location choices for adding your application on a BlackBerry device<br>• Media key event access<br>• Multiple contact lists support<br>• Phone screen customization<br>• Support for the BlackBerry Push Service on a BlackBerry device<br>• System requirements |
| 9 November 2009 | Added the following topics:<br>• Autocomplete text field<br>• Biometric recognition and authentication<br>• BlackBerry extension to JSR 179<br>• Date and time picker dialog boxes<br>• Enhanced Network API<br>• Fields that appear temporarily on a screen<br>• File picker<br>• Font customization<br>• Grid layouts<br>• Image creation<br>• Image scrolling |

| Date | Description |
| --- | --- |
| | • Image zooming and panning<br>• JavaScript extensions for a BlackBerry Widget application<br>• Landmark information retrieval using geocoding and reverse geocoding<br>• Location picker dialog boxes<br>• Location choices for adding your application on a BlackBerry device<br>• Message waiting indicator resets<br>• Spin box<br>• Support for the BlackBerry Push Service on a BlackBerry device<br>• Touch event injection<br>• User authentication modules<br><br>Changed the following topics:<br><br>• Screen transitions<br>• Overview |
| 6 October 2009 | Changed the following topics:<br><br>• Linking a contact<br>• MEID retrieval<br>• Multi-line phone support<br>• Phone screen customization<br>• Screen transitions |
| 14 August 2009 | Changed the System requirements topic. |

# Legal notice

8

©2010 Research In Motion Limited. All rights reserved. BlackBerry®, RIM®, Research In Motion®, SureType®, SurePress™ and related trademarks, names, and logos are the property of Research In Motion Limited and are registered and/or used in the U.S. and countries around the world.

Bluetooth is a trademark of Bluetooth SIG. Eclipse is a trademark of Eclipse Foundation, Inc. GSM is a trademark of the GSM MOU Association. Intel and Pentium are trademarks of Intel Corporation. Java, JavaScript, and NetBeans are trademarks of Sun Microsystems, Inc. OpenGL is a trademark of Silicon Graphics, Inc. SQLite is a trademark of Hipp, Wyrick & Company, Inc. Wi-Fi is a trademark of the Wi-Fi Alliance. All other trademarks are the property of their respective owners.

This documentation including all documentation incorporated by reference herein such as documentation provided or made available at www.blackberry.com/go/docs is provided or made accessible "AS IS" and "AS AVAILABLE" and without condition, endorsement, guarantee, representation, or warranty of any kind by Research In Motion Limited and its affiliated companies ("RIM") and RIM assumes no responsibility for any typographical, technical, or other inaccuracies, errors, or omissions in this documentation. In order to protect RIM proprietary and confidential information and/or trade secrets, this documentation may describe some aspects of RIM technology in generalized terms. RIM reserves the right to periodically change information that is contained in this documentation; however, RIM makes no commitment to provide any such changes, updates, enhancements, or other additions to this documentation to you in a timely manner or at all.

This documentation might contain references to third-party sources of information, hardware or software, products or services including components and content such as content protected by copyright and/or third-party web sites (collectively the "Third Party Products and Services"). RIM does not control, and is not responsible for, any Third Party Products and Services including, without limitation the content, accuracy, copyright compliance, compatibility, performance, trustworthiness, legality, decency, links, or any other aspect of Third Party Products and Services. The inclusion of a reference to Third Party Products and Services in this documentation does not imply endorsement by RIM of the Third Party Products and Services or the third party in any way.

EXCEPT TO THE EXTENT SPECIFICALLY PROHIBITED BY APPLICABLE LAW IN YOUR JURISDICTION, ALL CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS, OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY CONDITIONS, ENDORSEMENTS, GUARANTEES, REPRESENTATIONS OR WARRANTIES OF DURABILITY, FITNESS FOR A PARTICULAR PURPOSE OR USE, MERCHANTABILITY, MERCHANTABLE QUALITY, NON-INFRINGEMENT, SATISFACTORY QUALITY, OR TITLE, OR ARISING FROM A STATUTE OR CUSTOM OR A COURSE OF DEALING OR USAGE OF TRADE, OR RELATED TO THE DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN, ARE HEREBY EXCLUDED. YOU MAY ALSO HAVE OTHER RIGHTS THAT VARY BY STATE OR PROVINCE. SOME JURISDICTIONS MAY NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES AND CONDITIONS. TO THE EXTENT PERMITTED BY LAW, ANY IMPLIED WARRANTIES OR CONDITIONS RELATING TO THE DOCUMENTATION TO THE EXTENT THEY CANNOT BE EXCLUDED AS SET OUT ABOVE, BUT CAN BE LIMITED, ARE HEREBY LIMITED TO NINETY (90) DAYS FROM THE DATE YOU FIRST ACQUIRED THE DOCUMENTATION OR THE ITEM THAT IS THE SUBJECT OF THE CLAIM.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, IN NO EVENT SHALL RIM BE LIABLE FOR ANY TYPE OF DAMAGES RELATED TO THIS DOCUMENTATION OR ITS USE, OR PERFORMANCE OR NON-PERFORMANCE OF ANY SOFTWARE, HARDWARE, SERVICE, OR ANY THIRD PARTY PRODUCTS AND SERVICES REFERENCED HEREIN INCLUDING WITHOUT LIMITATION ANY OF THE FOLLOWING DAMAGES: DIRECT, CONSEQUENTIAL, EXEMPLARY,

INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR AGGRAVATED DAMAGES, DAMAGES FOR LOSS OF PROFITS OR REVENUES, FAILURE TO REALIZE ANY EXPECTED SAVINGS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF BUSINESS OPPORTUNITY, OR CORRUPTION OR LOSS OF DATA, FAILURES TO TRANSMIT OR RECEIVE ANY DATA, PROBLEMS ASSOCIATED WITH ANY APPLICATIONS USED IN CONJUNCTION WITH RIM PRODUCTS OR SERVICES, DOWNTIME COSTS, LOSS OF THE USE OF RIM PRODUCTS OR SERVICES OR ANY PORTION THEREOF OR OF ANY AIRTIME SERVICES, COST OF SUBSTITUTE GOODS, COSTS OF COVER, FACILITIES OR SERVICES, COST OF CAPITAL, OR OTHER SIMILAR PECUNIARY LOSSES, WHETHER OR NOT SUCH DAMAGES WERE FORESEEN OR UNFORESEEN, AND EVEN IF RIM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW IN YOUR JURISDICTION, RIM SHALL HAVE NO OTHER OBLIGATION, DUTY, OR LIABILITY WHATSOEVER IN CONTRACT, TORT, OR OTHERWISE TO YOU INCLUDING ANY LIABILITY FOR NEGLIGENCE OR STRICT LIABILITY.

THE LIMITATIONS, EXCLUSIONS, AND DISCLAIMERS HEREIN SHALL APPLY: (A) IRRESPECTIVE OF THE NATURE OF THE CAUSE OF ACTION, DEMAND, OR ACTION BY YOU INCLUDING BUT NOT LIMITED TO BREACH OF CONTRACT, NEGLIGENCE, TORT, STRICT LIABILITY OR ANY OTHER LEGAL THEORY AND SHALL SURVIVE A FUNDAMENTAL BREACH OR BREACHES OR THE FAILURE OF THE ESSENTIAL PURPOSE OF THIS AGREEMENT OR OF ANY REMEDY CONTAINED HEREIN; AND (B) TO RIM AND ITS AFFILIATED COMPANIES, THEIR SUCCESSORS, ASSIGNS, AGENTS, SUPPLIERS (INCLUDING AIRTIME SERVICE PROVIDERS), AUTHORIZED RIM DISTRIBUTORS (ALSO INCLUDING AIRTIME SERVICE PROVIDERS) AND THEIR RESPECTIVE DIRECTORS, EMPLOYEES, AND INDEPENDENT CONTRACTORS.

IN ADDITION TO THE LIMITATIONS AND EXCLUSIONS SET OUT ABOVE, IN NO EVENT SHALL ANY DIRECTOR, EMPLOYEE, AGENT, DISTRIBUTOR, SUPPLIER, INDEPENDENT CONTRACTOR OF RIM OR ANY AFFILIATES OF RIM HAVE ANY LIABILITY ARISING FROM OR RELATED TO THE DOCUMENTATION.

Prior to subscribing for, installing, or using any Third Party Products and Services, it is your responsibility to ensure that your airtime service provider has agreed to support all of their features. Some airtime service providers might not offer Internet browsing functionality with a subscription to the BlackBerry® Internet Service. Check with your service provider for availability, roaming arrangements, service plans and features. Installation or use of Third Party Products and Services with RIM's products and services may require one or more patent, trademark, copyright, or other licenses in order to avoid infringement or violation of third party rights. You are solely responsible for determining whether to use Third Party Products and Services and if any third party licenses are required to do so. If required you are responsible for acquiring them. You should not install or use Third Party Products and Services until all necessary licenses have been acquired. Any Third Party Products and Services that are provided with RIM's products and services are provided as a convenience to you and are provided "AS IS" with no express or implied conditions, endorsements, guarantees, representations, or warranties of any kind by RIM and RIM assumes no liability whatsoever, in relation thereto. Your use of Third Party Products and Services shall be governed by and subject to you agreeing to the terms of separate licenses and other agreements applicable thereto with third parties, except to the extent expressly covered by a license or other agreement with RIM.

Certain features outlined in this documentation require a minimum version of BlackBerry® Enterprise Server, BlackBerry® Desktop Software, and/or BlackBerry® Device Software.

The terms of use of any RIM product or service are set out in a separate license or other agreement with RIM applicable thereto. NOTHING IN THIS DOCUMENTATION IS INTENDED TO SUPERSEDE ANY EXPRESS WRITTEN AGREEMENTS OR WARRANTIES PROVIDED BY RIM FOR PORTIONS OF ANY RIM PRODUCT OR SERVICE OTHER THAN THIS DOCUMENTATION.

Research In Motion Limited
295 Phillip Street
Waterloo, ON N2L 3W8
Canada

Research In Motion UK Limited
Centrum House
36 Station Road
Egham, Surrey TW20 9LF
United Kingdom

Published in Canada